```
from sklearn import datasets
from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt
import numpy as np

iris = datasets.load_iris()
X = iris.data[0:100, 0:2]
Y = iris.target[0:100]

# Seperate the test data nad the train data
x_train, x_test, y_train, y_test = train_test_split(X, Y, test_size=0.8, random_state=0)
```

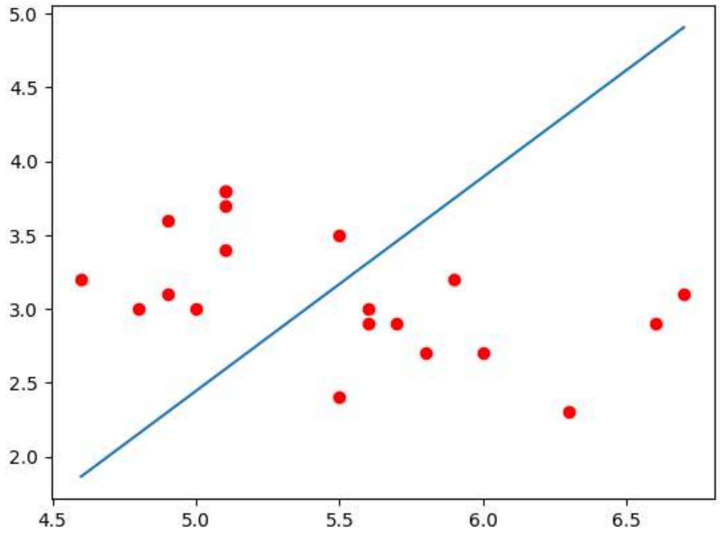
Question 1: Binary Linear Classifier On 2 Features

```
from sklearn.linear_model import LogisticRegression

model = LogisticRegression(random_state=0).fit(x_train, y_train)


decisionLineX = np.linspace(min(x_train[:, 0]), max(x_train[:, 0]))
decisionLineY = (- model.coef_.T[0] / model.coef_.T[1]) * decisionLineX + (- model.intercept_[0] / model.coef_.T[1])
for i in range(len(x_train)):
    plt.plot(x_train[i, 0], x_train[i, 1], "o", color ='red')
plt.plot(decisionLineX, decisionLineY)
```

```
[<matplotlib.lines.Line2D at 0x7a3223d4c880>]
```



Question 2: Accuracy

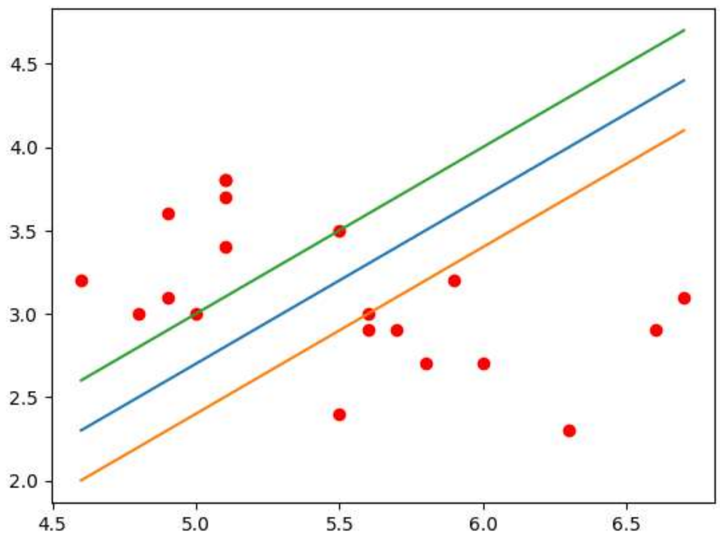
```
score = model.score(x_train, y_train)
print(f"Accuracy of model (Train): {score}")
score = model.score(x_test, y_test)
print(f"Accuracy of model (Test): {score}")
```

```
Accuracy of model (Train): 1.0
Accuracy of model (Test): 0.9875
```

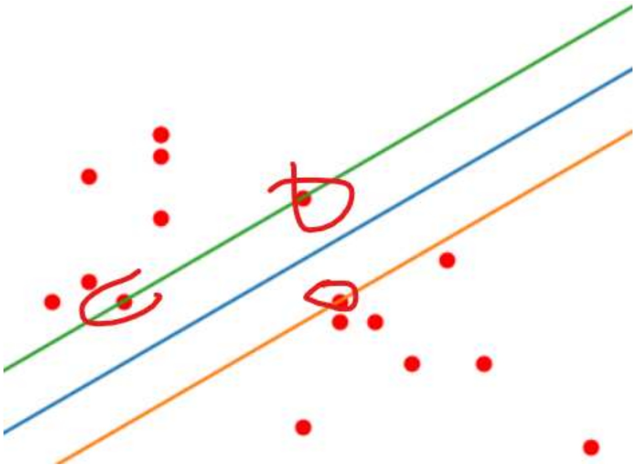
Question 3: SVM

```
from sklearn.svm import SVC
model = SVC(kernel='linear',random_state=0, C=100000).fit(x_train, y_train)


decisionLineX = np.linspace(min(x_train[:, 0]), max(x_train[:, 0]))
decisionLineY = (- model.coef_.T[0] / model.coef_.T[1]) * decisionLineX + (- model.intercept_[0] / model.coef_.T[1])
margin = 1 / np.sqrt(np.sum(model.coef_**2))
marginLineYup = decisionLineY - np.sqrt(1 + (- model.coef_.T[0] / model.coef_.T[1])**2) * margin
marginLineYdown = decisionLineY + np.sqrt(1 + (- model.coef_.T[0] / model.coef_.T[1])**2) * margin
for i in range(len(x_train)):
  plt.plot(x_train[i, 0], x_train[i,1], 'o', color='red')
plt.plot(decisionLineX, decisionLineY,decisionLineX, marginLineYup,decisionLineX, marginLineYdown)
plt.show()
```



Question 4: The support vectors are the points that are intersecting with the margin line.



Question 5: Accuracy

```
score = model.score(x_train, y_train)
print(f"Accuracy of model (Train): {score}")
score = model.score(x_test, y_test)
print(f"Accuracy of model (Test): {score}")
```

```
    Accuracy of model (Train): 1.0
    Accuracy of model (Test): 1.0
```

Question 6: Value of margin

```
print("Margin: "+ str(margin))
```

```
    Margin: 0.21215037731981287
```